

# Procédure d'installation et de déploiement

## Application Web de Gestion des Absences avec Django

---

### 1. Architecture de la solution

#### Infrastructure technique

##### VM1 - Serveur de base de données :

- Système : Debian 12
- SGBD : MariaDB 10.11
- Port : 3306

##### VM2 - Serveur d'application :

- Système : Debian 12
- Framework : Django 4.2.7
- Serveur WSGI : Gunicorn
- Serveur web : Nginx (reverse proxy)
- Port : 80/443

#### Modèle de données

```
groupe (id_gr, nom_gr)
├─ etudiants (id_etu, nom_etu, prenom_etu, mail_etu, groupe, photo)
└─ cours (id_cours, titre_cours, date_cours, enseignants, groupe, duree_cours)
    └─ absences (etudiants, cours, justifie, justification, img)

enseignants (id_ens, nom_ens, prenom_ens, mail_ens)
```

---

### 2. Installation

## VM1 - Serveur de base de données

```
# Installation MariaDB
sudo apt update && sudo apt upgrade -y
sudo apt install mariadb-server mariadb-client -y
sudo systemctl enable mariadb
sudo mysql_secure_installation

# Configuration réseau
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
# Modifier : bind-address = 0.0.0.0
sudo systemctl restart mariadb
```

## VM2 - Serveur d'application

```
# Installation des dépendances
sudo apt update && sudo apt upgrade -y
sudo apt install python3 python3-pip python3-venv nginx git -y
sudo apt install mariadb-client libmariadb-dev gcc python3-dev -y

# Création utilisateur
sudo useradd -m -s /bin/bash django_app
```

## Configuration Django

```
# Environnement virtuel
python3 -m venv venv
source venv/bin/activate
pip install django mysqlclient pillow python-decouple gunicorn

# Création projet
django-admin startproject gestion_absences
cd gestion_absences
python manage.py startapp absences
```

**Configuration base de données (settings.py) :**

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'gestion_absences',
        'USER': 'django_user',
        'PASSWORD': 'mot_de_passe',
        'HOST': 'adresse_ip_vml',
        'PORT': '3306',
    }
}
```

## 3. Déploiement

### Configuration Gunicorn

**Service systemd** (/etc/systemd/system/gunicorn.service):

```
[Unit]
Description=gunicorn daemon
After=network.target

[Service]
User=django_app
Group=www-data
WorkingDirectory=/home/django_app/gestion_absences
ExecStart=/home/django_app/gestion_absences/venv/bin/gunicorn \
    --workers 3 \
    --bind unix:/home/django_app/gestion_absences/app.sock \
    gestion_absences.wsgi:application

[Install]
WantedBy=multi-user.target
```

### Configuration Nginx

**Site configuration** (/etc/nginx/sites-available/gestion\_absences):

```
server {  
    listen 80;  
    server_name adresse_ip_vm2;  
  
    location /static/ {  
        root /home/django_app/gestion_absences;  
    }  
  
    location /media/ {  
        root /home/django_app/gestion_absences;  
    }  
  
    location / {  
        include proxy_params;  
        proxy_pass http://unix:/home/django_app/gestion_absences/app.sock;  
    }  
}
```

## Activation des services

```
# Activation  
sudo systemctl enable gunicorn  
sudo systemctl start gunicorn  
sudo systemctl enable nginx  
sudo systemctl restart nginx  
  
# Vérification  
sudo systemctl status gunicorn  
sudo systemctl status nginx
```

---

## 4. Tests de validation

```
# Test base de données  
mysql -h adresse_ip_vm1 -u django_user -p gestion_absences  
  
# Test application web  
curl http://adresse_ip_vm2/admin/
```

---

Cette architecture distribuée garantit une solution robuste et évolutive pour la gestion des absences, avec une séparation claire entre les couches de données et d'application.